DOCUMENT RESUME

ED 180 446 IR 007 920

AUTHOR Roberts, Stephen D.; And Others
TITLE Modeling and Simulation with INS.
PUB DATE 6 Apr 79
NOTE 19p.; Paper presented at the Annual Conference on Academic Computing Applications (Gary, IN, April 6, 1979); Figures may not reproduce; For related document, see IR 007 915

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Models; Networks; *Prediction; *Programing Languages; *Simulation; Systems Analysis
IDENTIFIERS *Integrated Network Simulation Language

ABSTRACT

INS, the Integrated Network Simulation language, puts simulation modeling into a network framework and automatically performs such programming activities as placing the problem into a next event structure, coding events, collecting statistics, monitoring status, and formatting reports. To do this, INS provides a set of symbols (nodes and branches) corresponding to familiar processes such as queues and activities which can be combined to create a network model of a system such as health care, manufacturing, and computer systems. The resulting network provides an excellent means of communicating the structure of the system to other people. The modeler translates the INS network into INS statements which are then read and executed by INS. INS automatically provides comprehensive statistical output on the behavior of the model. Based on the output, the original model can easily be altered and extended. Furthermore, experiments can be conducted in which characteristics of the network are changed. The result of these experiments can then be used to make predictions about the effects of making similar changes to the system being studied. INS is powerful yet easy to use and teach and requires no prior knowledge of simulation or programming. A textbook on INS, written in standard FORTRAN for portability, is available. (Author/JEG)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED180446

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

MODELING AND SIMULATION WITH INS

by

Stephen D. Roberts
James Schier
and
Thom Sulanke

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Stephen D. Roberts

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR007920

MODELING AND SIMULATION WITH INS

Stephen D. Roberts, Ph.D.
Director

James Schier
Programmer/Analyst

Thom Sulanke
Programmer/Analyst

Health Systems Research Section
Regenstrief Institute for Health Care
1001 West Tenth Street
Indianapolis, Indiana, 46202

ABSTRACT

Networks can be used to describe a large class of systems including health care, manufacturing, and computer systems. INS, the Integrated Network Simulation language, provides a set of symbols (nodes and branches) corresponding to familiar processes such as queues and activities which can be combined to create a network model of a system. The resulting network provides an excellent means of communicating the structure of the system to other people. The modeler translates the INS network into INS statements which are then read and executed by INS. INS automatically provides comprehensive statistical output on the behavior of the model. Based on the output, the original model can easily be altered and extended. Furthermore, experiments can be conducted in which characteristics of the network are changed. The result of these experiments can then be used to make predictions about the effects of making similar changes to the system being studied. INS is powerful yet easy to use and teach and requires no prior knowledge of simulation or programming. A textbook on INS is available. INS is written in standard FORTRAN for portability.

INTRODUCTION

Computer simulation is increasingly being used to aid decision-makers in a variety of environments including manufacturing, production, service, and public sectors. Building a simulation model can be greatly simplified by using a simulation language. For many languages, the simulation modeler must place the problem into a next event structure, code events, collect statistics, monitor status, and format reports. Hence the modeler must be a programmer who is also well founded in simulation concepts. In addition, the resultant simulation model will have limited meaning to anyone other than the model builder.

INS, the Integrated Network Simulation language, overcomes many of these deficiencies by putting simulation modeling into a network framework and automatically performing the programming activities mentioned above. The INS modeler conceives the simulation problem graphically as a network, composed of predefined modeling symbols (nodes and branches). These modeling symbols correspond to familiar processes such as queues and activities and they enable the modeler

to easily abstract directly from the real system to the INS network. The symbols are translated to INS statements. INS automatically executes the simulation and provides output. Consequently, the INS modeler can focus on the problems of modeling and not be impeded by programming tasks. INS requires no competence in computer programming.

Since INS is concise, easy to document, and visually appealing, INS models are also excellent vehicles for communication between modelers and their clients. Such communication encourages the decision-makers to actively participate in the modeling process, thus increasing the probability of eventual implementation.

The purpose of this paper is to illustrate the use of INS in model building and systems analysis. Space limits us to a discussion of a single example and a few embellishments. You are referred to the textbook (1) that fully describes the language and includes examples to illustrate its many modeling features.

BASIC CONCEPTS IN INS

The INS network is constructed about the flow of units of traffic called transactions. A transaction is a generic term which is interpreted by the modeler. For example, transactions may represent TVs coming into an inspection station, people arriving for hair cuts, ships entering a harbor, or customers coming into a gas station. The nodes and branches within the network depict creating transactions, routing them through the network, queuing at activities, servicing at activities, and eventually departing from the network.

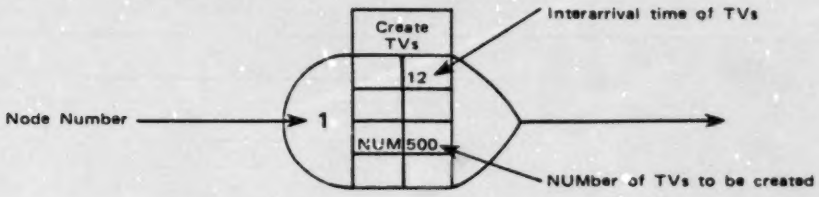
Transactions may require resources to process them in the network. The resource in INS is the generic term applied to an entity which responds to the demands of transactions within the network. In INS the resource can also exercise independent decision-making in fulfilling its service requirement. Examples of resources are inspectors who inspect TVs, barbers who cut hair, tugs which assist ships into a harbor, and gas station attendants and pumps which service customers.

A TV INSPECTION AND ADJUSTMENT PROBLEM

As a portion of their production process, TV sets are sent to a final inspection station at the rate of one set every 12 minutes. The inspector spends 10 minutes examining a TV. Of the sets inspected, 85% pass but 15% fail and are sent to an adjusting station. The adjustor repairs the set in 30 minutes and sends it back for reinspection where the set again has a 15% chance of failure.

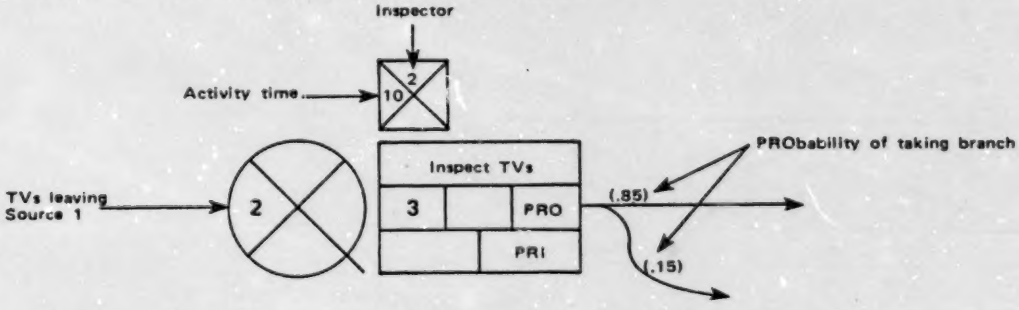
This problem is easily interpreted as a network. We begin by identifying the entities involved. Transactions will be used to represent TVs since they are the units of traffic demanding service in the system. The inspector and adjustor are resources since they

provide service to the TVs and constrain the rate at which TVs are processed. The TVs will enter the network at a node called a source node. The source node controls when and how many TVs will arrive. Let us say we want 500 TVs to be created at 12 minute intervals. The source node looks like the following:



Every node is given an identifying node number appearing on the left hand side of the node. The arrow pointing out of the node is called a branch and indicates the node to which the TVs must go next. In this manner, nodes are connected to form the network.

Inspection and adjustment of TVs are represented by activity nodes. Activities are places where transactions (TVs) are delayed, usually to receive service. The inspector and adjustor constrain the processing and no more than one TV at a time can be inspected or adjusted. Thus, other TVs are forced to wait for service in queue nodes. Queue and activity nodes are always adjacent to each other so that no branching is required. For the inspection activity, the queue and activity nodes look like this:



TVs wait in Queue 2 (shaped like a large Q) until the inspector (resource number 2) can inspect them. Following the 10 minute inspection at Activity 3, 85% of the TVs leave via one branch while 15% leave via the other branch heading for a similar queue and activity representing TV adjustment.

TVs which successfully pass inspection are no longer needed and they leave the network at a sink node:

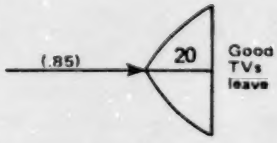
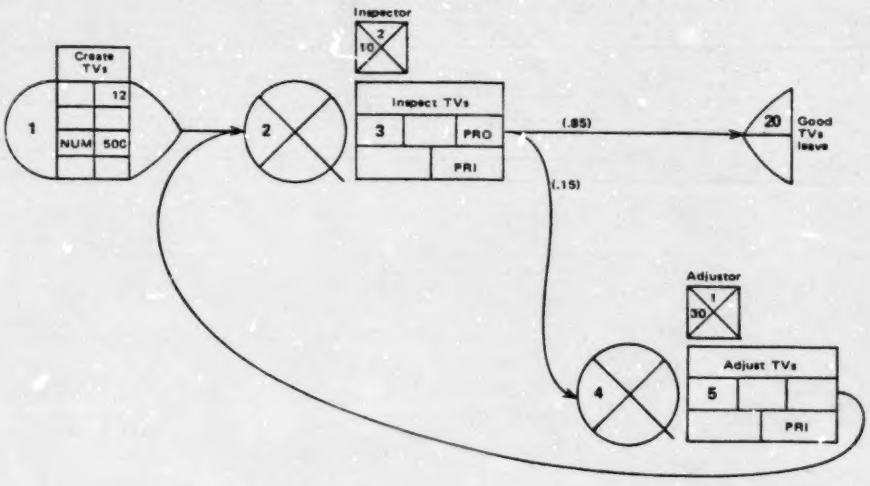


Figure 1 shows the nodes assembled to form the entire network.

Figure 1: Basic TV Inspection and Adjustment Problem



RESOURCES AVAILABLE	
Description	Number
Adjustor	1
Inspector	2

Figure 2: INS Statements for the Basic Network

```
 RESOURCE,1,1                          $ Adjustor
 RESOURCE,2,2                          $ Inspector
$
 SOURCE,1,,,12,,,NUMBER OF TVS =,500   $ Create TVs
     BRANCH TO, 2
$
 QUEUE,2                               $ Wait for inspector
 ACTIVITY,3,PRO,,,,,PRI                $ Inspect TVs
     GROUP,1, 2,,,10                   $ Using the inspector for 10 minutes
     BRANCH TO,20, .85                 $ 85% pass
     BRANCH TO, 4, .15                 $ 15% fail
$
 QUEUE,4                               $ Wait for adjustor
 ACTIVITY,5,,,,,,PRI                   $ Adjust TVs
     GROUP,1, 1,,,30                   $ Using the adjustor for 30 minutes
     BRANCH TO,2                       $ Return for re-inspection
$
 SINK,20                               $ Good TVs leave
```

We have used the nodes as building blocks, arranging them to agree with our conception of how the real system works. We have connected the nodes with branches to characterize the flow of transactions from node to node thereby allowing us to create models of any size or complexity out of simple, basic processes. Changes in individual processes are confined to single nodes. Furthermore, entire models can be easily altered by adding, subtracting, or rearranging the nodes and branches. The network is intuitively appealing since it corresponds visually to our understanding of the real system.

Before we demonstrate the ease of enhancing the basic model, let us see how our network is translated into a form suitable for the computer. Each node, branch, and resource in Figure 1 is translated to an INS statement by the direct correspondence of specifications on the node to fields on a statement. Fields in INS statements are separated by commas. Thus, the network in Figure 1 becomes the set of statements in Figure 2. The dollar sign ($) is used to indicate that no more fields exist, allowing comments to be added which help to document and clarify the network. The use of this feature makes the INS input very readable. The third field of the SOURCE and ACTIVITY statements is used to control the method of branching. TVs branch from Activity 3 by PRObability; at other nodes, TVs go directly to the indicated node. Other methods of branching (not shown) allow transactions to test system conditions such as queue length and current time. The GROUP statement corresponds to the square symbol above the activity node and tells INS which resources are capable of servicing transactions at the activity as well as the activity time. Multiple commas are used to skip unused fields corresponding to the blank specifications on the node.

Expanding the TV Inspection and Adjustment Problem

Let us now embellish our system and see how it affects the model. Suppose the inspection of TVs is divided into two parts. The first part requires an inspector to use a circuit testing machine while the second part requires only the inspector. Also assume a second, more experienced inspector is now available. This inspector can also adjust TVs but prefers to inspect them. When given a choice between inspecting a newly arrived TV and one which has been adjusted, this older inspector prefers to reinspect the adjusted TV. If both of the inspectors are available when a TV arrives to be inspected, the newer inspector should be selected leaving the more highly skilled inspector available for other tasks. Finally, the activity times for inspection and adjustment are no longer constant. It has been determined that the inspection time while using the testing machine is Normally distributed with a mean of 10 minutes and a standard deviation of 2 minutes. Inspection without the machine is Uniformly distributed between 2 and 8 minutes and adjustment time conforms to an Erlang distribution of order 2 with a mean of 45 minutes. The interarrival of TVs is Exponentially distributed with a mean of 12 minutes.

Figure 3 shows the modified network. The inspection activity has been divided into two nodes. Activity 3 represents the inspection with the testing machine and Activity 7 is the inspection without the testing machine. So that adjusted TVs can be logically distinguished from newly arrived TVs, Queue 6 has been added. Activity 3 now requires two resources, represented by the two columns of resource symbols. One resource from each column must be available before Activity 3 can begin. Symbols within one column specify alternative resources. In selecting an inspector at Activity 3, the newer inspector has 1st PRIority while the older inspector has 2nd priority. At the adjustment activity, the adjustor is given PRIority over the older inspector. At Activity 7 no queue is required because the TV simply retains the inspector from the CURrently completed inspection at Activity 3.

The older inspector may service TVs in Queues 2, 6, and 4. When the older inspector finishes an activity, TVs may be waiting in all three queues. Which one should be serviced? The decision is made by using a queue selector tree which describes the inspector's decision-making process. A selector node like node 30 is used to specify one decision. In this case, the older inspector PREfers to service TVs in Queue 6, then TVs in Queue 2, and lastly, TVs in Queue 4. The ability of resources to make decisions exerts a strong influence on the performance of a model and is a convenient modeling concept in INS.

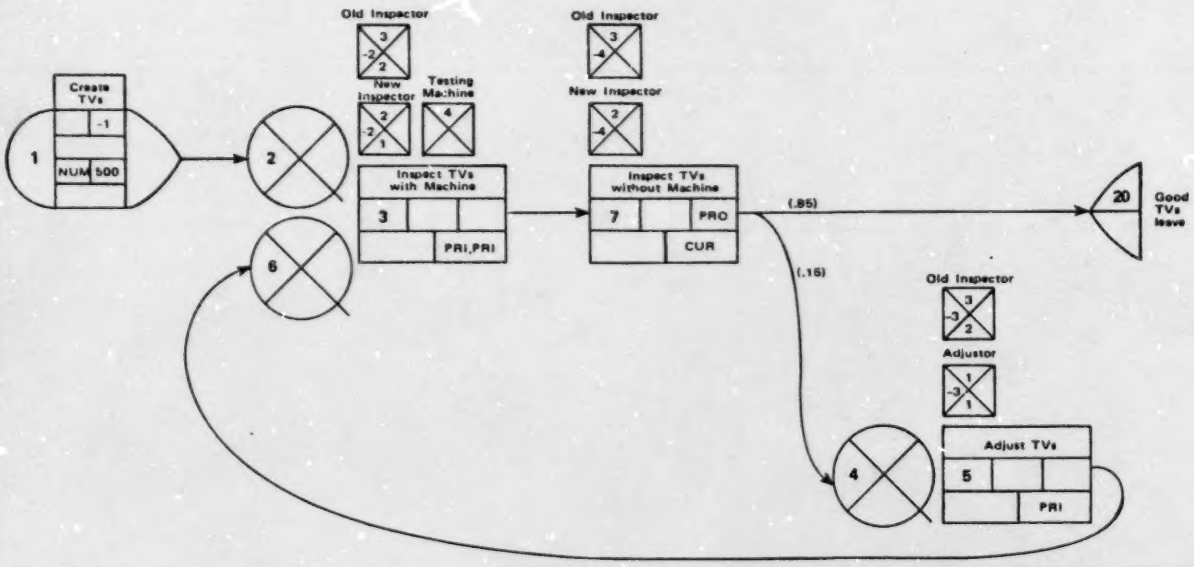
Finally, the constant times in the previous version have been replaced by negative numbers referring to parameter sets which specify random behavior according to a statistical distribution. INS automatically generates random deviates from a number of well known distributions and also allows the modeler to specify his or her own distribution.

Adding Individuality to the TVs

Transactions may not always represent undifferentiated units. We can distinguish transactions by assigning them attributes. For example, suppose there are two different types of TVs produced. Each type of TV requires a slightly different inspection time when using the circuit tester. Also, the testing machine is quite expensive and we wish to collect statistics on the utilization of the machine by the two types of TVs. The testing maching costs $300 per hour or $5 per minute. Finally, we wish to give TVs which have been adjusted more than once a higher priority at the inspection activity.

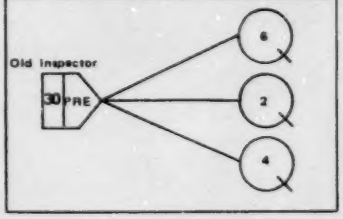
Figure 4 shows that a second source node has been added to model the second type of TV. TVs created at Source 1 are given the attribute TYPe = 1 while those created at Source 8 have TYPe = 2. A new node now appears called the assignment node. It has the following two parts:

Figure 3: Expanded TV Inspection and Adjustment Problem



RESOURCES AVAILABLE		
Description	Number	Selector Node
Adjustor	1	--
New Inspector	2	--
Old Inspector	3	30
Testing Machine	4	--

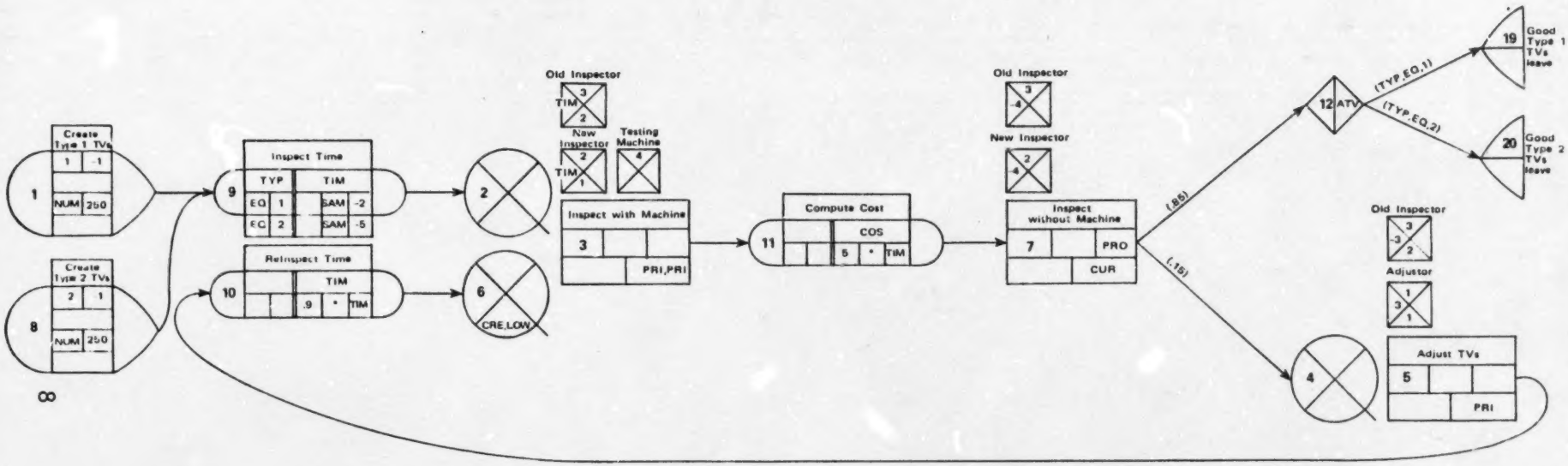
QUEUE SELECTOR TREE



PARAMETER SETS

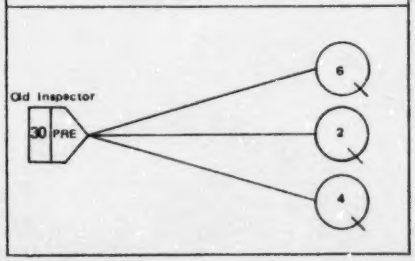
-1. Exponential interarrival time of TVs
-2. Normal inspection time with machine
-3. Erlang-2 adjustment time
-4. Uniform inspection time without tester

Figure 4: Adding Individuality to the TVs



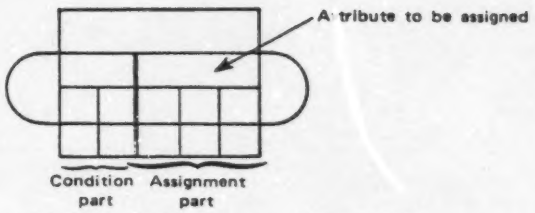
RESOURCES AVAILABLE		
Description	Number	Selector Node
Adjustor	1	–
New Inspector	2	–
Old Inspector	3	30
Testing Machine	4	–

QUEUE SELECTOR TREE



PARAMETER SETS

-1. Exponential interarrival time of TVs
-2. Normal inspection time with machine (TYP=1)
-3. Erlang-2 adjustment time
-4. Uniform inspection time without tester
-5. Normal inspection time with machine (TYP=2)



At Assignment 9, if TYPe EQuals 1, then TIMe is assigned a SAMple from Parameter set 2. If TYPe EQuals 2, then TIMe is assigned a SAMple from Parameter set 5. At Activity 3, we can see that TIMe appears in place of the -2, and thus INS will automatically use the TIMe obtained at Assignment 9.

Following inspection, we compute the COSt of using the machine as $5 per minute times the inspection TIMe (in minutes). By adding a TABLE statement (which can be seen in Figure 8), we can create a table containing the following information on machine cost: total cost; average cost per inspection; standard deviation, minimum, and maximum cost per inspection; and all of these statistics for each type of TV separately. Figure 5 contains the table produced on cost. The average cost per inspection was $53.97 while the total cost of using the machine was $7016.25. After being adjusted, inspection time is recomputed as a fraction of the original time. At Queue 6, adjusted TVs are now ranked by LOWest CREation time, an attribute automatically assigned by INS when a transaction is created. This differs from the default ranking of first-in-first-out (FIFO) at Queues 2 and 4. A TV which has been adjusted more than once will be older than the other TVs and will be placed at the front of Queue 6.

Finally, good TVs enter a route node which enables transactions to make more than one decision in choosing a final destination. At Route 12, the TYPe 1 TVs branch to Sink 19 while TYPe 2 TVs branch to Sink 20. This allows us to collect statistics at the sink nodes for each type of TV.

Enlarging the Scope of the Model

In our final enhancement, we wish to add to our model the following processes. After being successfully inspected, TVs are placed in boxes by an automated boxing machine. The boxer is subject to periodic breakdowns about every 100 minutes, Exponentially distributed. When the boxer breaks down, it is repaired while inspected TVs must wait for repairs to be completed. The repair requires a Lognoramlly distributed time having a mean of 20 and a standard deviation of 5 minutes. After being boxed, TVs are moved to a pallet. When the pallet has eight TVs, it is moved into a temporary storage area.

Figure 6 shows the additions made. At Activity 13, TVs are boxed and then moved to a pallet at Queue 14. At Queue 14, TVs are GATHERed until 8 have accumulated. Then all 8 TVs are stored

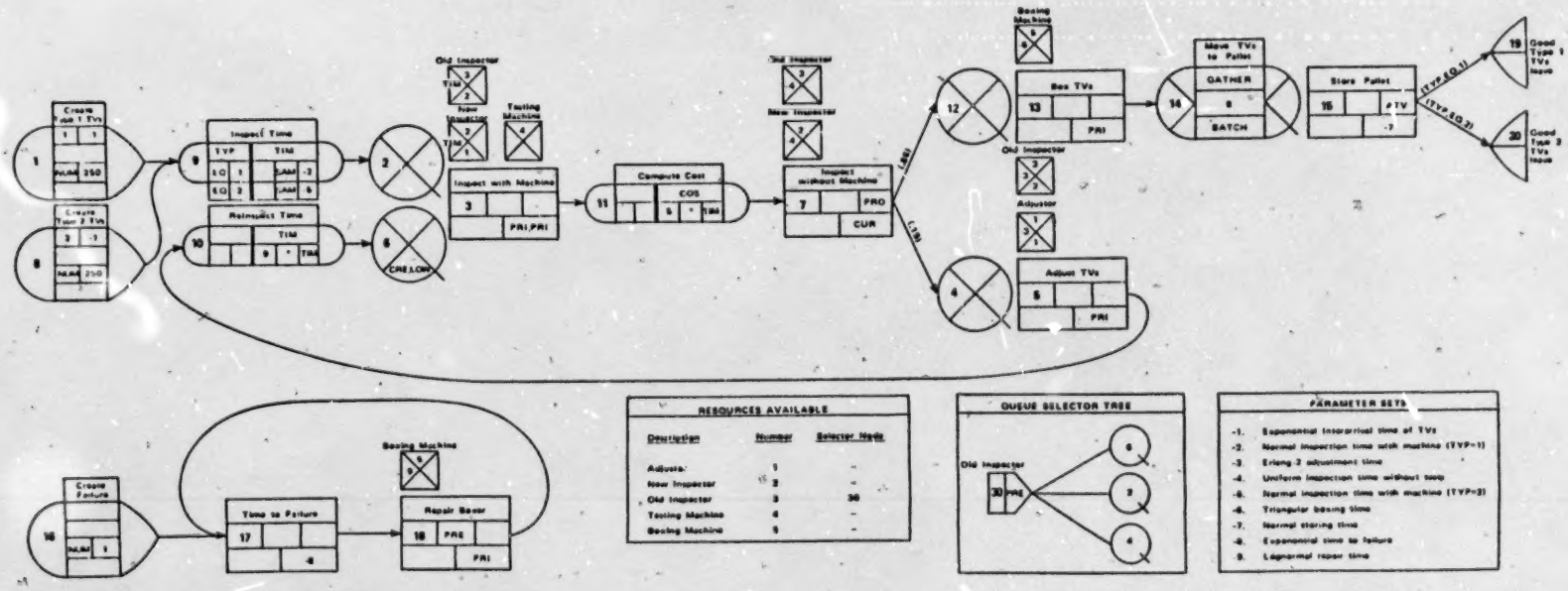
Figure 5: Table of Testing Machine's Cost

** TABLE NUMBER 1 **

COST OF TESTING MACHINE

TRANSACTION TYPE	MEAN	STANDARD DEVIATION	MINIMUM	MAXIMUM	NUMBER OF OBSERVATIONS	SUM	SUM OF THE SQUARES
1.00000	58.6484	15.8902	25.0000	100.000	62	3636.20	228660.
2.00000	49.7066	11.5102	23.4000	75.0000	68	3380.05	176887.
TOTAL	53.9712	14.4329	23.4000	100.000	130	7016.25	405547.

Figure 6: Boxing and Storing TVs



together as a BATCH. Although no resources are required at Activity 15, Queue 14 is necessary because transactions must wait for other transactions. This process is known as synchronization and gathering transactions is only one of several synchronization mechanisms available in INS.

To model the breakdown of the boxer, we add a second network. One transaction is created representing the failure of the boxer. After spending an Exponential amount of time at Activity 17, it enters Activity 18 which PREempts the boxing machine from Activity 13. The TV being boxed at Activity 13 is suspended until repairs are completed on the boxer (i.e. until Activity 18 ends). At that time, the boxer resumes boxing the TV and the failure transaction returns for another period of waiting at Activity 17.

INS OUTPUT

In a few pages we have presented a broad cross section of the modeling concepts available in INS. We have taken an elementary problem and expanded its scope in several directions with little difficulty. In addition to the ease with which modeling in INS is accomplished, INS automatically collects and reports a comprehensive set of statistics on all aspects of the model's behavior. Reports are produced on the number and time of transactions in the network, the number and time of transactions in all queues and activities, and the utilization of each resource. Also provided is information to assist the modeler in verifying the correct behavior of the network. For example, INS keeps track of the number of times each node is entered by transactions. A node which has never been entered frequently indicates an error in branching. The modeler can also request a trace of the transactions and resources moving through the network. Figure 7 shows part of the trace (generated by the MONITOR statement seen in Figure 8 which contains the entire set of statements needed to run the final version of the problem).

The statistics produced automatically by INS can be eliminated or enhanced by the modeler. We have already seen how INS can collect statistics on attributes. Additional statistics can also be collected on the nodes. Table 2 in Figure 8 collects statistics on the number of transactions in Queues 2 and 6 and Activities 3 and 7, i.e. the entire inspection subsystem. Figure 9 contains the histogram produced as part of Table 2. The all limits correspond to the number of TVs in the inspection area. We can see, for example, that there are 10 or fewer TVs in the area 79.3% of the time. Other forms of statistics are also created just as easily providing the user with an ensemble of output capable of verifying the network's behavior, providing answers to questions about performance, productivity, and cost, and generating additional insight into the system being studied.

APPLICATION AND AVAILABILITY OF INS

INS has been applied in several environments. The 37 examples in the textbook (1) illustrate the range of applications for INS

Figure 7: Tracing Transactions and Resources

TIME	ENTITY	NUMBER	(TYPE)	ACTION		NODE TYPE	NUMBER	RELATED INFORMATION		
381,350	TRN	23	(2,00)	COMPLETED	AT	ACTIVITY	3	ACTIVITY TIME =	16,640	
	RES	3	(2)	IN BUFFER STORAGE						
	RES	4	(3)	IN BUFFER STORAGE						
	TRN	23	(2,02)	LEAVING		ASSIGNMENT	11	INDIVIDUAL ATTRIBUTE 005 =	81.2400	
	RES	3	(2)	BUSY	AT	ACTIVITY	7	ON TRANSACTION 23	WITH A RES ACTIVITY	
	TRN	23	(2,00)	BEGINNING	AT	ACTIVITY	7			
	TRN	24	(2,00)	DEPARTING		QUEUE	2	TIME IN QUEUE =	36,470	
	RES	4	(3)	BUSY	AT	ACTIVITY	3	ON TRANSACTION 24	WITH A RES ACTIVITY	
	RES	2	(2)	BUSY	AT	ACTIVITY	3	ON TRANSACTION 24	WITH A RES ACTIVITY	
	TRN	24	(2,00)	BEGINNING	AT	ACTIVITY	3			
389,270	TRN	11	(2,00)	COMPLETED	AT	ACTIVITY	15	ACTIVITY TIME =	55,650	
				DESTROYED	AT	SINK	20			
	TRN	12	(1,02)	COMPLETED	AT	ACTIVITY	15	ACTIVITY TIME =	55,650	
				DESTROYED	AT	SINK	19			
	TRN	13	(1,02)	COMPLETED	AT	ACTIVITY	15	ACTIVITY TIME =	55,650	
				DESTROYED	AT	SINK	19			
	TRN	15	(1,00)	COMPLETED	AT	ACTIVITY	15	ACTIVITY TIME =	55,650	
				DESTROYED	AT	SINK	19			
	TRN	16	(2,00)	COMPLETED	AT	ACTIVITY	15	ACTIVITY TIME =	55,650	
				DESTROYED	AT	SINK	20			
	TRN	9	(2,00)	COMPLETED	AT	ACTIVITY	15	ACTIVITY TIME =	55,650	
				DESTROYED	AT	SINK	20			
	TRN	14	(2,00)	COMPLETED	AT	ACTIVITY	15	ACTIVITY TIME =	55,650	
				DESTROYED	AT	SINK	20			
	TRN	17	(1,00)	COMPLETED	AT	ACTIVITY	15	ACTIVITY TIME =	55,650	
				DESTROYED	AT	SINK	19			
389,670	TRN	29	(1,00)	CREATED	AT	SOURCE	1	APPOINTMENT TIME =	389,670	
				LEAVING		ASSIGNMENT	9	INDIVIDUAL ATTRIBUTE 11 =	12.5600	
				ENTERING		QUEUE	2	THE NUMBER OF TRANSACTIONS WAITING IS		4
389,750	TRN	23	(2,00)	COMPLETED	AT	ACTIVITY	7	ACTIVITY TIME =	5,480	
	RES	3	(2)	IN BUFFER STORAGE						
	TRN	23	(2,00)	ENTERING		QUEUE	12	THE NUMBER OF TRANSACTIONS WAITING IS		1
	RES	3	(2)	IDLE						
387,990	TRN	21	(1,00)	COMPLETED	AT	ACTIVITY	13	ACTIVITY TIME =	16,210	
	RES	5	(4)	IN BUFFER STORAGE						
	TRN	21	(1,00)	ENTERING		QUEUE	14	THE NUMBER OF TRANSACTIONS WAITING IS		3
				GROUPED	AT	QUEUE	14	WITH TRANSACTION 18		
	TRN	22	(1,00)	DEPARTING		QUEUE	12	TIME IN QUEUE =	10,610	
	RES	5	(4)	BUSY	AT	ACTIVITY	13	ON TRANSACTION 22	WITH A RES ACTIVITY	
	TRN	22	(1,00)	BEGINNING	AT	ACTIVITY	13			

Figure 8: Statements for the Final Version

```
$                FINAL TV INSPECTION AND ADJUSTMENT PROBLEM
$
 GENERAL, INSPECT AND ADJUST TVS,1,(7)TIME TO STOP =,1500 $ MINUTES
 SET,(8)2
$          SET DISTRIBUTION  MEAN  MIN  MAX
 PARAMETER, 1, EXPONENTIAL ,,  23                  $ TV INTERARRIVAL TIME
 PARAMETER, 2,   NORMAL    ,,  12,  5,  20, 3 $ INSPECT TIME W/ MACHINE--TYPE 1
 PARAMETER, 3,   ERLANG    ,,  45, 30,  70, 2 $ ADJUSTMENT TIME
 PARAMETER, 4,   UNIFORM   ,,    ,  2,   8    $ INSPECT TIME WITHOUT MACHINE
 PARAMETER, 5,   NORMAL    ,,  10,  5,  15, 2 $ INSPECT TIME W/ MACHINE--TYPE 2
 PARAMETER, 6, TRIANGULAR  ,,  10,  8,  12    $ BOXING TIME
 PARAMETER, 7,   NORMAL    ,,  50, 30,  70, 5 $ STORAGE TIME
 PARAMETER, 8, EXPONENTIAL ,, 100,             $ TIME TO FAILURE
 PARAMETER, 9,  LOGNORMAL  ,,  20,  5,  50, 5 $ REPAIR TIME
$
 TABLE, 1, COST OF TESTING MACHINE, A/T
 TABLE, 2, TVS IN INSPECTION AREA,, 15, 1, 1
$
 MONITOR,TRACE AT TIME ,300
 MONITOR,UNTRACE AT TIME ,330
$
 ATTRIBUTE,TIME                          $ INSPECT TIME WITH MACHINE
 ATTRIBUTE,COST,(6)1                     $ COST OF TESTING MACHINE - TABLE 1
$
$        NUMBER  TYPE  SELECTOR
 RESOURCE,   1,    1                     $ ADJUSTOR
 RESOURCE,   2,    2                     $ NEW INSPECTOR
 RESOURCE,   3,    2,     30             $ OLD INSPECTOR
 RESOURCE,   4,    3                     $ TESTING MACHINE
 RESOURCE,   5,    4                     $ BOXING MACHINE
$
 SELECTOR TREE FOR OLD INSPECTOR, 30, PREFERRED ORDER, 6, 2, 4
$
$******************************** THE NETWORK ********************************
$
 SOURCE,1,,1,-1,,,NUMBER TO CREATE =,250 $ CREATE TYPE 1 TV SETS
    BRANCH TO, 9
 SOURCE,8,,2,-1,,,NUMBER TO CREATE =,250 $ CREATE TYPE 2 TV SETS
$
 ASSIGNMENT, 9,, TYPE ,       TIME =   ,
                 EQ, 1,    , SAMPLE, -2,
                 EQ, 2,    , SAMPLE, -5  $ ASSIGN INSPECTION TIME BY TV TYPE
$
 QUEUE,2,2                               $ QUEUE OF ARRIVING TVS
 QUEUE,6,2,(6)CREATION TIME,LOWEST FIRST $ QUEUE OF ADJUSTED TVS
 ACTIVITY,3,,2,(8)PRIORITY,,PRIORITY     $ INSPECT TVS WITH MACHINE
    GROUP,1, 2,,1,TIME, 3,,2,TIME        $ BY AN INSPECTOR
    GROUP,2, 4                           $ USING THE TESTING MACHINE
$
 ASSIGNMENT,11,,,COST =,,,5,*,TIME       $ COMPUTE COST AT $5 PER MINUTE
 ACTIVITY,7,,PRO,2,(8)CURRENT            $ INSPECT TVS WITHOUT MACHINE
    GROUP,1, 2,,,-4, 3,,,-4              $ BY THE SAME INSPECTOR
    BRANCH TO, 12, .85                   $ 85 PERCENT PASS
    BRANCH TO,  4, .15                   $ 15 PERCENT FAIL
$
 QUEUE,4
 ACTIVITY,5,(8)PRIORITY                  $ ADJUST TVS
    GROUP,1, 1,,1,-3, 3,,2,-3            $ BY THE ADJUSTOR OR OLD INSPECTOR
 ASSIGNMENT,10,,,TIME =,,, .9,*,TIM      $ ASSIGN REINSPECTION TIME
    BRANCH TO, 6                         $ RETURN FOR REINSPECTION
$
 QUEUE,12
 ACTIVITY,13,(8)PRIORITY                 $ BOX TVS
    GROUP,1, 5,,,-6                      $ BY THE BOXING MACHINE
$
 QUEUE,14                                $ MOVE TVS TO PALLET
    SYNCHRONIZE,GATHER,8,BATCH           $ WITH 8 TVS PER PALLET
 ACTIVITY,15,ATV,(8)-7                   $ STORE PALLET
    BRANCH TO, 19, TYPE,EQ,1
    BRANCH TO, 20, TYPE,EQ,2
$
 SINK,19                                 $ GOOD TYPE 1 TVS LEAVE
 SINK,20                                 $ GOOD TYPE 2 TVS LEAVE
$
$************** BOXING MACHINE FAILURE LOOP **************
$
 SOURCE,16,,1                            $ CREATE FAILURE
$
 ACTIVITY,17,(8)-8                       $ EXPONENTIAL TIME TO NEXT FAILURE
$
 ACTIVITY,18,(6)PREEMPT,,PRIORITY        $ REPAIR BOXING MACHINE
    GROUP,1, 5,,,-9                      $ USING THE BOXING MACHINE
    BRANCH TO, 17                        $ LOOP BACK FOR MORE FAILURES
FINISH
```

Figure 9: Distribution of the Number of TVs in the Inspection Area

** TABLE NUMBER 2 **

TVS IN INSPECTION AREA

TRANSACTION TYPE	MEAN	STANDARD DEVIATION	MINIMUM	MAXIMUM	TIME OF OBSERVATION	SUM	SUM OF THE SQUARES
TOTAL	6,44254	3,90915	,000000E+00	14,0000	1551,00	9668,81	84524.1

** HISTOGRAM NUMBER 2 **

TVS IN INSPECTION AREA

TIME OF OBSERVATION	RELATIVE FREQUENCY	CUMULATIVE FREQUENCY	---CELL LIMIT--- LOWER	UPPER	0,0 ... 0,2 ... 0,4 ... 0,6 ... 0,8 ... 1,0
9,19200	2,026	2,026	-INFINITY	5,[illegible]E+00	+
86,7300	2,258	4,764	0,000000E+00	1,0000	+**C
173,000	4,116	8,184	1,0000	2,0000	+******* C
133,200	6,289	6,269	2,0000	3,0000	+***** C
187,370	7,125	0,394	3,0000	4,0000	+******* C
130,630	5,092	4,486	4,0000	5,0000	+****** C
117,770	2,779	2,565	5,0000	6,0000	+***** C
88,3600	2,059	0,625	6,0000	7,0000	+**** C
87,9320	2,059	2,462	7,0000	8,0000	+**** C
95,1200	2,203	0,745	8,0000	9,0000	+**** C
71,3800	1,648	6,793	9,0000	10,000	+*** C
92,6100	2,162	0,955	10,000	11,000	+**** C
93,6100	2,162	1,117	11,000	12,000	+**** C
92,2600	2,151	2,979	12,000	13,000	+**** C
32,1200	0,721	1,700	13,000	14,000	+* C
,000000E+00	0,000	1,700	14,000	15,000	+ C
,000000E+00	0,000	1,700	15,000	+INFINITY	+ C
				UPPER LIMIT INCLUSIVE	0,0 ... 0,2 ... 0,4 ... 0,6 ... 0,8 ... 1,0

including queuing, inventory, computer, manufacturing, and service systems.

INS has been successfully used in simulation courses for engineering students. Graduate students utilized it as a new network-based language while it has been used as the principle simulation methodology in an undergraduate industrial engineering course in systems analysis. The language has also been used as the basis of a senior elective in medical school. Medical students have used INS to create models for clinical investigation. These instances of its use in a teaching environment have demonstrated that, not only can the language be quickly learned and applied by persons with minimal technical expertise, but it can also be valuable to those with deeper appreciation of simulation.

INS is written in ANSI FORTRAN making it portable and has been implemented on a number of machines. INS and the textbook are available from the Regenstrief Institute, 1001 West Tenth Street, Indianapolis, Indiana, 46202. The text, Simulation Modeling With INS, is available for $15. INS is actively being maintained by the Institute and is the subject of continuing improvement both to increase the efficiency of its execution and to extend its applications. For further information you can call (317) 264-4566.

CONCLUSION

Because simulation modeling is a process of successive refinement, there is a need to be able to quickly produce a working "first cut" model and to embellish this model as greater insight is generated. INS facilitates model building by providing a simple, but powerful set of symbols which can be quickly combined to create a network model of a system. By a simple translation of the INS network to INS statements, INS conducts the simulation. It relieves the modeler from the burdens of simulation mechanics and programming. The approach allows the modeler to focus on the important tasks of model creation and validation. INS makes embellishments to the model easy because they can be accomplished by changing a specification at a node or altering some segment of the network. INS is easy to learn and use, not only by making it easy to create a simulation model but by giving structure to the modeling process. Since INS is visually appealing and easily documented, it is an excellent vehicle for communication between the modeler and the client, encouraging cooperative participation. Last but not least, INS is written in ANSI standard FORTRAN and is available for a variety of computers.

REFERENCES

1. Roberts, S.D., and J.S. Schier, Simulation Modeling With INS, Regenstrief Institute for Health Care, Indianapolis, Indiana, 1979.